GRiDA: a Green Distributed Algorithm for Backbone Networks

Aruna Prem Bianzino[a,b], Luca Chiaraviglio[a], Marco Mellia[a]
a) Electronics Department, Politecnico di Torino, Torino, Italy
b) Institut TELECOM, TELECOM ParisTech, Paris, France
Email: {bianzino, chiaraviglio, mellia}@tlc.polito.it

***Abstract*—In this work, we face the problem of reducing the power consumption of Internet backbone networks. We propose a novel algorithm, called GRiDA, to selectively switch off links in an Internet Service Provider IP-based network to reduce the system energy consumption. Differently from approaches that have been proposed in the literature, our solution is completely distributed among the nodes. It leverages link state protocol like OSPF to limit the amount of shared information, and to reduce the algorithm complexity. Moreover, GRiDA does not require the knowledge of the actual traffic matrix, an unrealistic assumption common to all other proposals. Results, obtained on realistic case studies, show that GRiDA achieves performance comparable to several existing centralized algorithms.**

I. Introduction

According to different studies [1], [2], the carbon footprint of Information and Communication Technologies (ICT) is constantly increasing, representing today up to 10% of the global CO_2 emissions. Among the main ICT sectors, 37% of the total ICT emissions are due to telecommunication infrastructures and their devices, while data centers and user devices are responsible for the remaining part [1]. It is therefore not surprising that researchers, manufacturers and network providers are spending significant efforts to reduce the power consumption of ICT systems from different angles.

To this extent, networking devices waste a considerable amount of power. In particular, energy consumption has always been increased in the last years, coupled with the increase of the offered performance [3]. Actually, power consumption of networking devices scales with the installed capacity, rather than the current load [4]. Thus, for an Internet Service provider (ISP) the network power consumption is practically constant, unrespectively to traffic fluctuations, since all devices consumes always the same amount of power. In turn, devices are underutilized, especially during off-peak hours when traffic is low. This represents a clear opportunity for saving energy, since many resources (i.e., routers and links) are powered on without being fully utilized, while a carefully selected subset of them can be switched off without affecting the offered Quality of Service (QoS).

In the literature, different approaches have been proposed to reduce the gap between the capacity *offered* by the network and the resources *required* by users (see [3] and [5] for an overview). The proposed approaches can be divided into two main categories: *power proportional* techniques that adapt the capacity (and thus consumption) of the devices to the actual load, and *sleep mode* approaches, that leverage on the idea of introducing idle mode capabilities. While the first approach involves deep modifications in the design of hardware components, the second approach requires coordination among networking devices to carefully distribute the extra load that results from putting into sleep mode some devices.

In this paper, we face the problem of reducing power consumption in backbone networks adopting a sleep mode approach. The intuition has been already proposed in the literature, starting from the seminal work of Gupta *et al.* [6]. In particular, approaches ranging from traffic engineering [7], to routing protocols [8], and new architectures [9] have been proposed. These works tackle the minimization of network power consumption by powering off elements, such as routers and links, and large savings are possible when sleep mode states are exploited. However, to the best of our knowledge all of the previous work either assume the complete knowledge of the traffic matrix at each given time [7], [10], [11], or do not consider the traffic flowing in the network [8]. Similarly, all the previous solutions are completely centralized [10] or require at least the presence of a control node [9]. Thus, the applicability of the aforementioned approaches is limited to specific cases.

In our work, we follow a different approach: we propose a novel distributed algorithm, called GRiDA, to put into sleep mode links in an IP-based network. Our solution is distributed among the nodes to (i) limit the amount of shared information, (ii) avoid explicit coordination among nodes, and (iii) reduce the problem complexity. Contrary to previous works, we assume that nodes do not know the traffic matrix, whose knowledge is indeed unrealistic in the current Internet architecture. Thus, the switch off decision is taken considering the current load of links *and* the history of past decisions. Thanks to the use of the history, our solution reduces the number of link reconfigurations to ease routing protocols convergence. GRiDA is able to react both to traffic variations and link/node failures.

We assess the effectiveness of our solution on realistic case studies and real topologies. Results show that GRiDA achieves performance comparable with the centralized solutions that assume the perfect knowledge of the traffic matrix.

The paper is organized as follows: the description of the algorithm is reported in Sec. II. Sec. III describes the realistic case studies considered for algorithm evaluation. Results are presented in Sec. IV. Finally, conclusions are drawn in Sec. V.

II. Algorithm Description

The GRiDA algorithm aims at reducing the network power consumption by adapting the network capacity to current traffic demand. In particular, it (i) switches off links when they are underutilized, and their absence in the network does not affect the network functionalities, and (ii) switches on idle links when capacity is required to guarantee a proper reaction to faults and changes in the traffic demand. The process of link switching off/on is decentralized to each node, which takes local decisions at random intervals without any coordination among the nodes.

We assume local decisions to be based only on the local node knowledge of the current load and power consumption of incident links, and on the knowledge of the current network topology, assured by a link-state routing algorithm, e.g., by OSPF or IS-IS. We assume nodes not to know the network traffic matrix, contrary to what usually hypothesized by other works in the literature.

We assume that Link-State Advertisement (LSA) messages distribute information about the current network topology, augmented by information about eventual congestion in the network, i.e., link load overcoming a threshold, or presence of disconnected source/destination pairs. They are delivered to nodes at fix time intervals (Δ_{LSA}), selected by the network administrator.

We represent the network infrastructure as a di-graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. Vertices represent network nodes, while edges represent network links, being $N = |V|$ and $L = |E|$ the number of nodes and links respectively.

A. The Node Choice

A decision of a node n corresponds to entering a specific node *configuration* $K_n \in \mathcal{K}_n$, where $\mathcal{K}_n$ is the set of all possible configurations for node n; a configuration K_n is a combination of on/off states for incident links. More formally, given a node n, of degree $d^{(n)}$, and an ordered list of the incident links (in lexicographical order), a configuration is the vector $(k_1^{(n)}, \ldots, k_d^{(n)})$ of the configurations of the $d^{(n)}$ incident links. The configuration $k_l^{(n)}$ of a link l is a binary variable indicating the state of the link ($k_l^{(n)} = 0$ if the link is powered off, and $k_l^{(n)} = 1$ if the link is powered on). Therefore $|\mathcal{K}_n| = 2^{d^{(n)}}$.

The *status* S_n of a node n is the vector $(s_1^{(n)}, \ldots, s_d^{(n)})$ of the status associated to all the $d^{(n)}$ links incident to n. For each link l the status $s_l^{(n)}$ may assume 3 possible values, defined on the bases of the load of the link (ρ) and a load threshold (ϕ). They are summarized in Tab. I.

A *utility* function is defined as: $U(K_n, S_n) = c(K_n) + p(K_n, S_n)$, where $c(K_n)$ is the power consumption of node n computed as the sum of the power of the on-links in configuration K_n, and $p(K_n, S_n)$ is a penalty associated to the configuration on the basis of the *status* and the *history*.

TABLE I
Link Status.

Status	Name	Description
$\rho = 0$	*off*	link powered off or not used
$0 < \rho \leq \phi$	*normal*	link used but not congested
$\rho > \phi$	*overloaded*	link congested

```
Node Choice
Input: $K^{\text{old}}$, $S$
Output: $K$, $K^{\text{old}}$, $S^{\text{old}}$
  $S^{\text{old}} = S$
  if lastLSA == OK:
    $K^\star = min_K \ U(K,S)$
    if (check_connectivity($K^\star$) == OK):
      $K = K^\star$
      if $K \neq K^{\text{old}}$:
        to_be_checked = TRUE
    else
        $p(K^\star,S) = p(K^\star,S) + \beta$
  else:
    $K$ = all_on configuration
```

Alg. 1: The pseudo-code of the *node choice* event.

Since the same procedure is applied to all nodes, from now on we get rid of the index n for ease of notation.

For a single node, the problem turns into selecting the best configuration that minimizes the power consumption, while guaranteeing the global system to work properly. This problem can be solved by the support of the *Q-learning* technique [12], as the node choice is a function of the current state of the same node, and each possible choice is associated to an estimated utility function, updated by learning. Hence, node decisions, in normal network working state (i.e., last LSA did not report anomalies) correspond to the K minimizing $U(K, S)$. To ensure *fast reaction* to faults and sudden traffic changes, we introduced three safety mechanisms:

- if a choice would lead to a network disconnection, it is not applied and its penalty is updated with an additive factor β as if a violation occurs (detailed in Sec.II-B);
- if a choice taken in a non congested network state is followed by a congestion reported by a LSA, the choice is regretted, i.e., the node returns to the previous configuration;
- in a congestion network state, a node which is taking a decision will automatically select the *all-on* state. This choice can not be regretted.

The pseudo code resuming the decision process is reported in Alg. 1, where S is the current state of the node.

B. The Penalty Evolution

The values of $p(K, S)$ are updated step-by-step, on the basis of the *history*: if the decision of entering configuration K when in status S is followed by an LSA reporting a network critical state, the cost associated to that choice (i.e., $p(K, S)$)

```
LSA Arrival
Input: $K$, $K^{\text{old}}$, $S^{\text{old}}$, $p$
Output: $K$, $p$
  if to_be_checked == TRUE:
    if LSA == OK:
      for $J$ in $\mathcal{K}$:
        $p(J,S^{\text{old}})$ = $p(J,S^{\text{old}})$ $\times$ $\delta$
    else:
      $p(K,S^{\text{old}})$ = $p(K,S^{\text{old}})$ + $\beta$
      $K$ = $K^{\text{old}}$
    to_be_checked = FALSE
```

Alg. 2: The pseudo-code of the *LSA arrival* event.

is incremented by an additive factor β (≥ 0):

$$p(K,S) = p(K,S) + \beta \tag{1}$$

If a decision is taken in state S and no violation is reported by the successive LSA, the costs associated to choices in state S (i.e., $p(*,S)$) are decremented by a multiplicative factor $\delta \leq 1$:

$$p(J,S) = p(J,S) \times \delta \qquad \forall J \in \mathcal{K}_n \tag{2}$$

Intuitively, (1) penalizes choices which likely brought to a violation of connectivity or capacity constraints; (2) pushes nodes toward *exploration* of all the possible choices by reducing the effect of the accumulated memory.

Penalty updates are performed when the LSA is received (except the choices that would lead to disconnection that are immediately penalized). The pseudo code describing them is reported in Alg. 2, where K is the current node configuration, K^{old} is the node configuration before the last choice, S^{old} is the node status at the time the last choice has been taken, and p is the penalty state of the node.

C. The Algorithm Initialization

In order to speed up convergence, the cost function $p(K,S)$ must be properly initialized. The intuition is to discriminate between (i) switching off an unloaded link (ii) switching off a link which is carrying traffic (iii) switching off a congested link. In addition, we need to avoid multiple attempts of radical switching off choices during convergence by further penalizing configurations with an higher number of off links and link loads larger than zero.

More formally, an initial penalty function $\theta_l(k_l, s_l)$ is associated to each configuration K and each possible status S:

$$\theta_l(k_l,s_l) = \begin{cases} 0 & s_l = off \ \vee k_l = 1 \\ 1/d & s_l = normal \ \wedge k_l = 0 \\ \varepsilon/d & s_l = overloaded \ \wedge k_l = 0 \end{cases} \tag{3}$$

Where ε is a constant ≥ 1. The $\frac{1}{d}$ factor is a normalization over the node degree.

Then, the penalty $p(K,S)$ is initialized to $\sum_{l \in n} \theta_l(k_l, s_l)$, with $k_l \in K$, $s_l \in S$. The procedure is repeated for all nodes $n \in V$, and for all configurations $K \in \mathcal{K}$ and all status $S \in \mathcal{S}$.

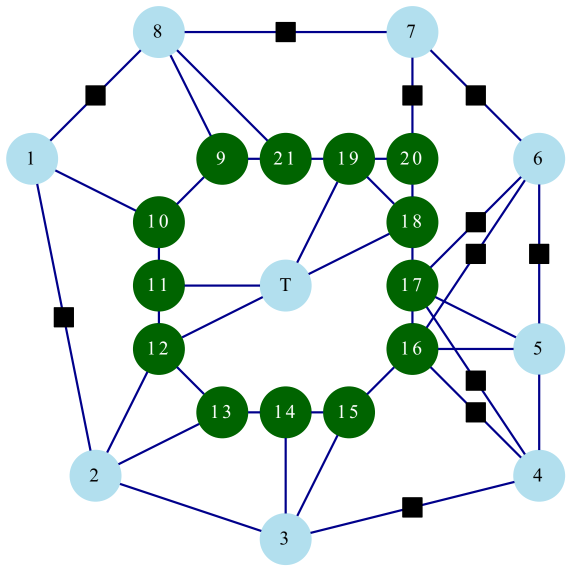


Fig. 1. A network topology from a telecom operator: ISP 1.

III. Scenario Description

To provide a relevant evaluation of the described algorithm, we tested it over 3 different scenarios, ranging from a metropolitan segment network to a European-wide network.

A. The Power and Traffic Model

In this work, we are interested in the power consumption related to links, i.e., the power consumption of the linecards, and of the optical amplifiers along the link. To have comparable results, we adopted here the same power model used in [10]. In particular, we consider ports consuming P_{nic} = 50 W for each c_{ref} = 10 Gbps of link capacity, and amplifiers consuming P_a = 1 kW for each c_{ref} = 10 Gbps of link capacity, with an amplifier every m_a = 70 km. Therefore, we compute the power consumption P_l of a link l, with capacity c_l and length m_l, as: $P_l = \lceil \frac{c_l}{c_{ref}} \rceil (\lfloor \frac{m_l}{m_a} \rfloor P_a + 2P_{nic})$.

In our simulations, we considered constant traffic requests over fixed time intervals Δ_{TM}, after which a new traffic matrix is considered. Traffic is expected to change on moderate time scale, so that $\Delta_{TM} = 30min$ or higher. The traffic matrices have been obtained from direct traffic measurements where available; otherwise, they are computed starting from a single measured traffic matrix and imposing an artificial traffic profile.

B. The Network Scenarios

ISP 1: The first testing scenario is an access/metropolitan segment of a traditional telecom operator network [13]. The topology is reported in Fig. 1, where nodes are represented by circles. Labels represent node IDs. This topology includes *access nodes* (IDs 1 to 8), which are sources and destinations of traffic requests, *transit nodes* (IDs 9 to 21), performing only traffic switching, and a *peering node* (ID *T*), providing access to the ISP transport network and the Internet. The small black squares in Fig. 1 indicate the presence of amplifiers on links.

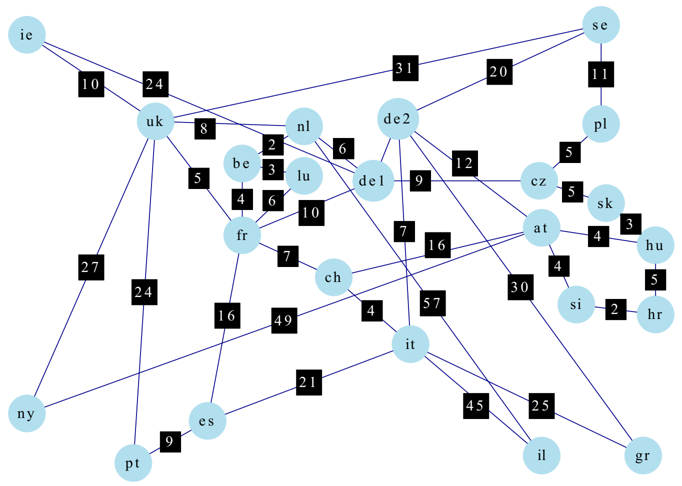


Fig. 2. The Geant network topology.

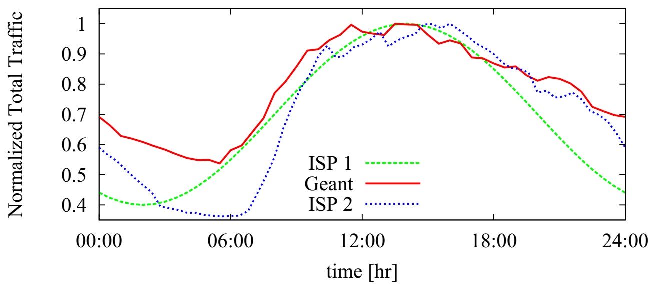


Fig. 3. Variation of the total traffic load versus time, normalized to the peak total traffic.

For this scenario, an actual traffic matrix has been provided. To add generality, we further consider a set of synthetic traffic matrices that assumes uniform traffic exchanged among access nodes and the traffic collection point. The maximum link utilization is guaranteed to be smaller than 70% ($\phi = 0.7$), and 47 traffic matrices have been generated applying the sinusoidal traffic profile described in [10], and represented in Fig. 3 by the green dashed line labeled "ISP 1"[1].

Geant: We consider the actual Geant Network [14], whose topology is reported in Fig. 2. Nodes are represented by circles, while black squares indicate the presence of optical amplifiers, whose number is reported as label. All nodes are sources and destinations of traffic. For this network topology, actual traffic matrices are publicly available, among which we selected the 48 traffic matrices of 05/05/2005 (a typical working day). The corresponding variation in terms of total traffic load is reported in Fig. 3 by the red continuous line.

ISP 2: Finally, we considered a topology inspired by the national network of an ISP (see [10] for details). It is a hierarchical network composed of 373 nodes, organized in 5 levels: core, backbone, metro, access and Internet nodes. The *core* level is composed by few nodes densely interconnected

[1]This is a national ISP network, where all nodes are in the same timezone.

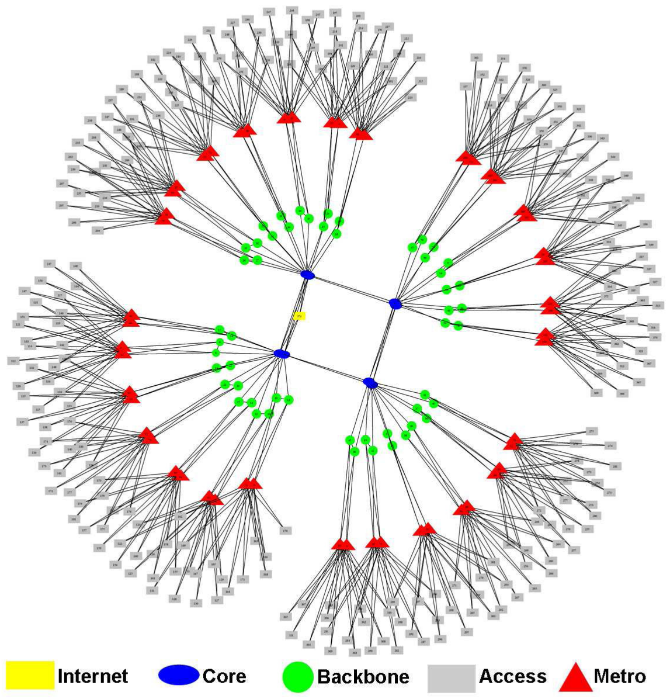


Fig. 4. A network topology from a telecom operator: ISP 2.

TABLE II
SIMULATION PARAMETERS IN THE 3 SIMULATION SCENARIOS.

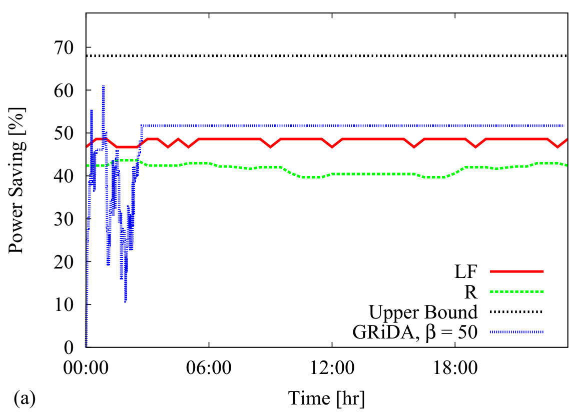
Parameter	ISP 1	Geant	ISP 2
Δ_{LSA} [s]	5	5	2
Δ_{TM} [min]	30	30	48
$\Delta_{c,Max}$ [s]	25	25	9
N	22	23	112 + 261
δ	1.0	0.999	0.999
β	50	50	100
ε	50	50	50
ϕ	0.7	0.7	0.5
Choices / Node / Traffic Matrix	5.5	5.2	4.7

by high-capacity links, and offering connectivity to the Internet by means of a peering node. Going down in the hierarchical levels, the number of nodes increases, and the link capacity decreases.

The access nodes and the Internet peering node are sources and destinations of traffic. The traffic requests for this topology have been generated following a measured traffic profile (reported in Fig. 3 by the blue dotted line), as described in [10][1] .

C. Parameter Setting

A new TM is considered every time interval Δ_{TM}. A randomly selected node is waken up to take a decision every random interval Δ_c, uniformly distributed between Δ_{LSA} and $\Delta_{c,Max}$ seconds. Time intervals must be chosen in order to have, on the one hand, at least one LSA occurrence between two consecutive decisions, and on the other hand, a significant number of decisions per node to allow algorithm convergence. On average, a single node takes a decision every $\Delta_c \times N$, where N is the number of nodes in the network.



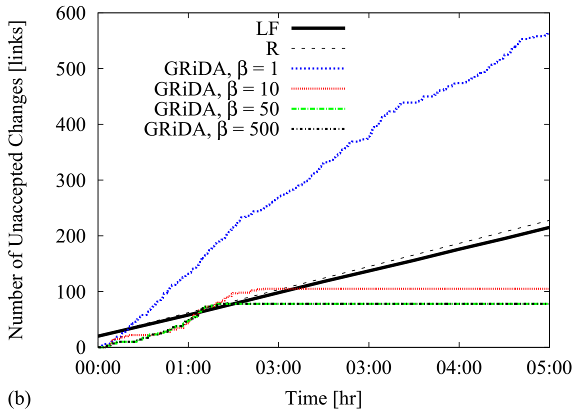


Fig. 5. ISP 1 network: (a) Power saving versus time, considering different algorithms, (b) cumulative number of unaccepted changes.

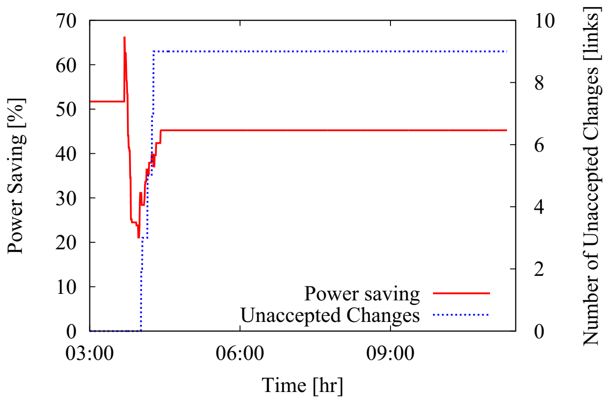


Fig. 6. ISP 1 network: Power saving and cumulative count of unaccepted changes with a fault occurring after convergence is reached.

Values for the parameters in the different simulation scenarios are summarized in Tab. II. The number of nodes for the *ISP 2* network is divided into two parts, as the first part (i.e., core, backbone, metro nodes) is running the GRiDA algorithm, while, the second part (i.e., access and Internet nodes) is not running the GRiDA algorithm. Access nodes in the *ISP 2* network are not directly connected among them, hence, every link is considered in the GRiDA algorithm even if they are not running it.

IV. Performance Evaluation

We start by evaluating the performance of GRiDA on the ISP 1 scenario. Unless otherwise specified, we use the parameters set of Tab. II. In particular, we start setting $\delta = 1$ for testing the convergence of the algorithm. We then compare the power saving of GRiDA against the upper bound obtained solving the optimal problem of [15] for the off-peak traffic, and the centralized Least-Flow (LF) and Random (R) heuristics of [11], which are heuristics that find the subset of links that must be powered off to carry the current traffic. In the two heuristics, links are firstly sorted by incremental carried traffic or in random order, respectively. The algorithms then iterates through the link list trying to see if it is possible to turn them off. In particular, for each given link, the link is turned off. Then traffic is routed over the residual capacity. If network connectivity and maximum link utilization constraints are met, link is definitively powered off. Otherwise the link is left in on state. A perfect knowledge of the traffic matrix is assumed to route traffic on the residual network and check connectivity constraints.

A. Transient Analysis and Parameter Sensitivity

Fig.5(a) reports the power-saving versus time of GRiDA, LF, R and the upper bound. It reports the power saving computed as the percentage of saved power with respect to a configuration in which all links are powered on. Since the LF and R heuristics are centralized and require the knowledge of the traffic matrix, we run them at every traffic matrix change. After an initial transient, the power saving of GRiDA is constant: this is due to the fact that $\delta = 1$ and the network is largely over-provisioned; thus the algorithm converges to a solution that does not involve any increment in the penalty function. Interestingly, GRiDA outperforms both the LF and R heuristics, saving 52% of power after convergence.

To give more insight, Fig.5(b) reports the cumulative number of link reconfigurations due to network violations, comparing LF, R and GRiDA for different values of β. LF and R are actually centralized solutions, which do not entail network violations. To obtain this metric for the two heuristics, we hence consider the number of times in which a link is considered in the heuristic, and is left on, due to load or connectivity constraint violations. Both LF and R show an increasing trend, since both regenerate a new solution at every run, resulting in a high number of (possible) violations. For what concerns GRiDA, reconfigurations occur only during the initial transient. To this extent, low values of β result in a large number of reconfigurations, since the learning rate of the algorithm is lower. The intuition suggests that in this case the predominant term in the utility function is the power consumption, thus each node always selects the most aggressive configuration in term of power savings, resulting in a large number of violations. On

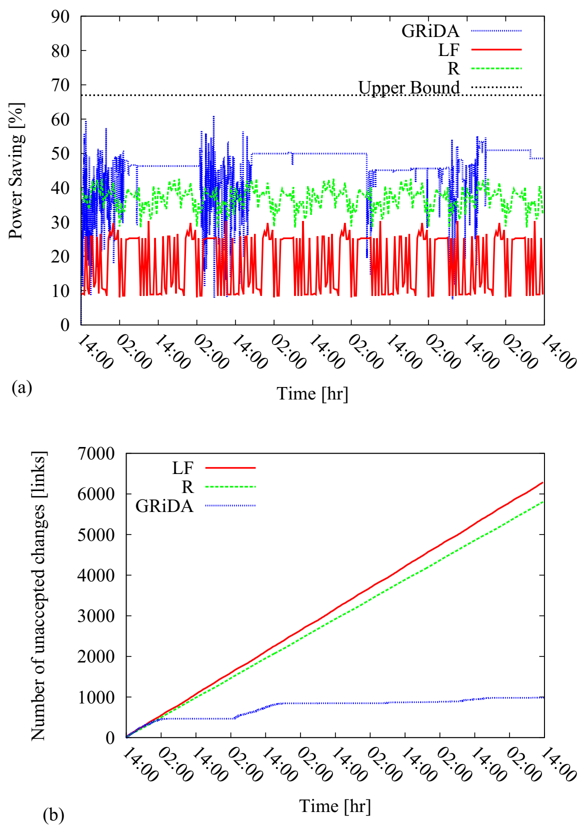


Fig. 7. Geant network: (a) Variation of power saving versus time, (b) Cumulative number of unaccepted changes.

the contrary, the number of reconfigurations steadily decreases, being far below the centralized heuristics. Thus, a trade off emerges among responsiveness of the algorithm and number of reconfigurations.

A similar sensitivity analysis has been performed over other algorithm parameters (i.e., δ, and ε), for all considered network scenarios, which is not reported here for lack of space. The resulting optimal values are reported in Tab. II.

We have evaluated the performance of GRiDA under anomalous network conditions. In particular, a link failure is simulated after convergence of GRiDA. Fig.6 reports both the power saving and the cumulative number of link reconfigurations before and after the failure event. GRiDA is able to wisely adapt to a new configuration with a limited number of reconfigurations. In fact, as soon the failure is detected GRiDA starts turning on links as long as LSA reports network anomalies. Then, the algorithm starts again to switch off links until a stable configuration is reached. While GRiDA has not been designed to explicitly handle failures, it helps the failure management algorithm to recover from critical conditions.

B. Experiments with Complex Networks

We consider now the Geant topology. Fig. 7(a) reports the power saving versus time. Also in this case GRiDA outperforms both the LF and R heuristics. Notice that here we have set $\delta = 0.999$, thus GRiDA does not converge to a stable solution; instead, it adapts the power saving to the actual traffic. Interestingly, the number of reconfigurations is still much lower than LF and R, as reported in Fig. 7(b).

We consider now the ISP 2 topology. In this case, we have taken as reference the Most Power (MP-MP) and Least Flow (LF-LF) heuristics, which has been proven in [10] to be the most effective ones for this topology. In particular, both MP-MP and LF-LF try to switch off first all the links incident to a node (which are sorted according to a Most Power or Least Flow order, respectively). Then, as a second step, the remaining links are eventually powered off individually (according to a Most Power or Least Flow ordering). We refer the reader to [10] for a detailed description of these algorithms.

Fig. 8(a) reports the algorithm comparison in terms of power saving. Interestingly, for all algorithms savings present a strong day-night trend. In particular, more power saving is possible when the network is lightly loaded, i.e., during night. In this case, GRiDA is able to save an amount of power comparable to centralized heuristics, but without requiring the knowledge of the current traffic matrix. Moreover, the variability of the traffic impose GRiDA to quickly adapt the configurations. To give more insight, Fig. 8(b) reports the average link load in the network running GRiDA. Results are averaged for each traffic matrix. The average load is below 10% during the day and about 20% during night, suggesting that the connectivity constraint is most likely faced during the night, while the capacity constraint is most likely predominant during the day.

The number of configurations resulting in $\rho > \phi$ is reported in Fig. 8(c) ($\phi = 0.5$). Results are again averaged for each traffic matrix. Here we report two events: link overloading ($\rho > 1$) and link congestion ($\phi < \rho \leq 1$). While the first event represent a potential issue for ISPs, the last one can be less critical. Interestingly, no violation occurs during the night, confirming the fact that the tightest constraint is to guarantee connectivity among sources and destinations. Thus, high power savings are possible without incurring in traffic violations. On the contrary, during the day violations occur. However, in this case the predominant event is link congestion, which is quickly recovered by reverting to less aggressive configurations.

Finally, Fig. 8(d) reports the average number of OFF-ON and ON-OFF link choices per each node. Note that here we are accounting also the link reconfigurations triggered by a negative LSA. The figure reports also the average node degree $\frac{L}{N}$. Interestingly, GRiDA tries to turn off on average one link per node every δ_{TM} during the night, being able to detect the lower traffic period. On the contrary, during the day GRiDA selects less aggressive strategies, i.e., on average, half link is switched off per node and δ_{TM}. Thus, we can conclude that GRiDA is very effective in switching off links in the network by following the traffic pattern. Moreover, GRiDA requires a really low number of reconfigurations, limiting hence the impact on the network functioning, and making the switch-on power spike negligible.

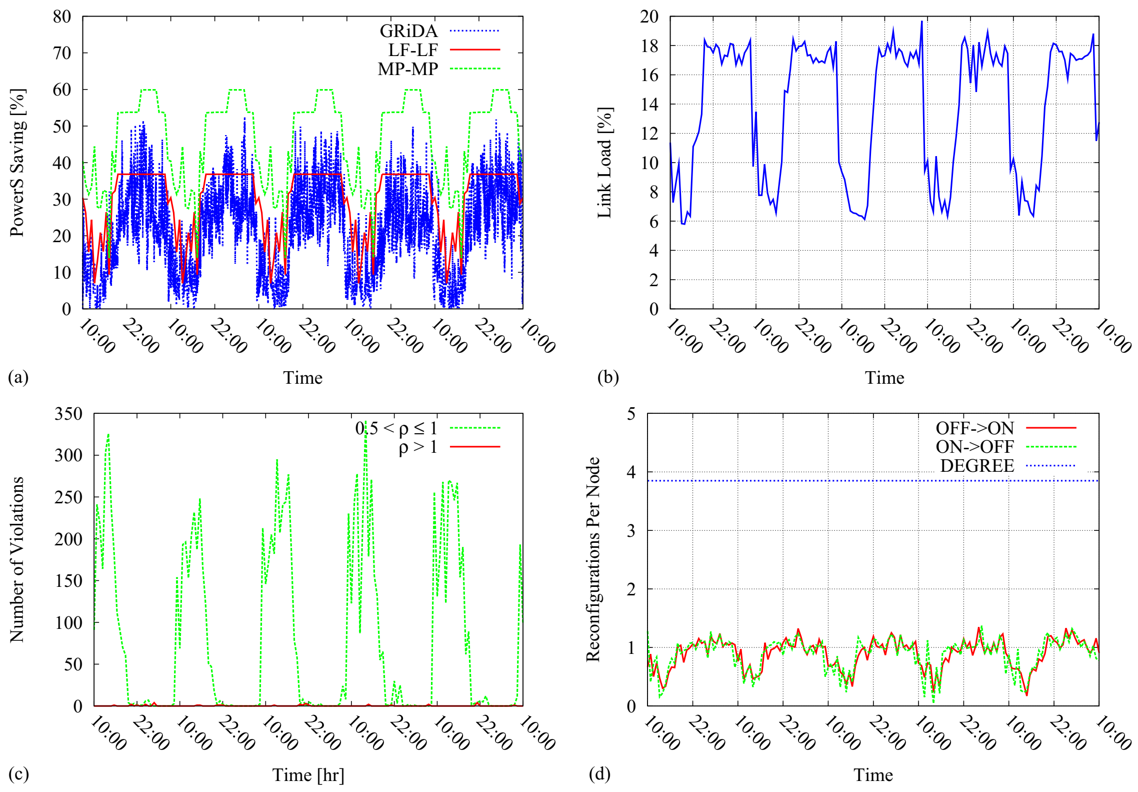


Fig. 8. ISP 2 network: (a) algorithms comparison, (b) average link load, (c) number of violations per Δ_{TM}, (d) OFF⇒ON and ON⇒OFF events per Δ_{TM} per node.

V. Conclusions

We have presented GRiDA, a distributed algorithm to reduce power consumption in backbone networks. Our solution is based on a reinforcement learning technique that requires only the exchange of periodic Link State Advertisements in the network. Results, obtained on realistic case studies, show that GRiDA achieves performance comparable to different existing algorithms.

As next steps, we will extend the power model adopted by considering the possibility of turning off full nodes rather than single links. Then, we plan to use local network topology information in the configuration choice to push further the algorithm performance. Finally, we will consider the impact of asynchronous timings in the exchange of information.

Acknowledgment

The work described in this paper was performed with the support of the ECONET project (low Energy COnsumption NETworks), funded by the EU through the FP7 call.

References

[1] M. Webb, "SMART 2020: Enabling the Low Carbon Economy in the Information Age." The Climate Group. London, June 2008.

[2] Global Action Plan, "An Inefficient Truth." Global Action Plan Report, http://globalactionplan.org.uk, December 2007.

[3] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures," *IEEE Communication Surveys and Tutorials*, 2011.

[4] A. Adelin, P. Owezarski, and T. Gayraud, "On the Impact of Monitoring Router Energy Consumption for Greening the Internet," in *IEEE/ACM International Conference on Grid Computing (Grid 2010)*, (Bruxelles, Belgique), October 2010.

[5] A. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, "A Survey of Green Networking Research," *IEEE Communication Surveys and Tutorials*, 2012.

[6] M. Gupta and S. Singh, "Greening of the Internet," in *ACM SIGCOMM 2003*, (Karlsrhue, Germany), August 2003.

[7] N. Vasić and D. Kostić, "Energy-aware traffic engineering," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking (e-Energy 2010)*, (Passau, Germany), April 2010.

[8] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini, "An Energy Saving Routing Algorithm for a Green OSPF Protocol," in *IEEE INFOCOM Workshops, 2010*, (San Diego, USA), March 2010.

[9] K. Ho and C. Cheung, "Green distributed routing protocol for sleep coordination in wired core networks," in *IEEE 6th International Conference on Networked Computing*, (Gyeongju, Korea (South)), May 2010.

[10] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware backbone networks: a case study," in *First Int. Workshop on Green Communications (GreenComm09)*, (Dresden, Germany), June 2009.

[11] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *IEEE ICC'09*, (Dresden, Germany), June 2009.

[12] C. Watkins and P. Dayan, "Q-learning," *MACHINE LEARNING*, vol. 8, no. 3, pp. 279–292, 1992.

[13] A. Bianzino, C. Chaudet, S. Moretti, D. Rossi, and J. Rougier, "The Green-Game: Striking a Balance between QoS and Energy Saving," in *Proceedings of the 23rd International Teletraffic Congress (ITC 2011)*, (San Francisco, USA), September 2011.

[14] The GEANT network, "http://www.geant.net/."

[15] A. Bianzino, C. Chaudet, F. Larroca, D. Rossi, and J. Rougier, "Energy-Aware Routing: a Reality Check," in *3rd International Workshop on Green Communications (GreenComm3)*, (Miami, USA), December 2010.